

C++ 프로그래밍 및 실습

University couple

진척 보고서 #2

제출일자 : 2023.11.26

제출자명 : 서현호

제출자학번 : 203566

1. 대학생들의 삶의 질 향상 및 추억 제공

1) 배경 및 필요성

코로나로 인해 대학생활을 하면서 즐길 수 있는 여러가지 활동들을 못하게 되었고 취업난으로 인해 자격증과 공부에 몰입할 수밖에 없는 구조에서 학생들은 인생의 한 번 뿐인 대학생활을 즐기지 못하고 삭막한 사회로 들어갈 위기에 처해져 있음. 특히 공대의 경우 남학생들의 비율이 압도적으로 많고 또 교육과 같은 경우에도 여학생들의 비율이 많음. 이러한 학생들은 연애를 시도하기조차 어렵고 비교적 이성을 만나기 쉬운 학생들도 연애에 더욱 신중해질 수밖에 없다. 그렇게 되면서 점점 캠퍼스의 낭만을 즐길 여유가 사라지고 있다. 그리하여 학생들에게 연애 할 기회를 주기 위해서 프로그램을 계획하게 되었음.

2) 프로젝트 목표

학생들의 성격과 연애관을 바탕으로 모집된 표본들 사이에서 일정 수준 이상의 연애관 일치가 존재하거나, 학생들이 좋아하는 성격을 서로 갖고 있다면 매칭을 해주어 이제 곧 사회 진출하기 직전의 전남대학교 학생들에게 캠퍼스의 낭만을 즐기게 해줄 수 있는 기회를 주는 것이 목표.

3) 차별점

기존의 매칭 프로그램들은 단순히 사진을 게시하고 간단한 채팅으로 서로를 알아 보는, 가벼운 만남을 할 수 밖에 없는 구조를 가진 프로그램들만 존재한다. 또한 불특정 다수의 알 수 없는 사람들이 모두 참여하는 프로그램과 달리 전남대학생 들을 대상으로 하였고 성격과 연애관을 중심으로 보아 기존의 가벼운 만남에서 탈피한 진중한 연애로 발전할 가능성이 더 높은 매칭 프로그램이다

2. 기능 계획

1) 기능 1 - 프로그램에 참여한 학생들의 정보를 입력 받기

(1) 세부 기능 1

- 학생들의 개인정보(인스타 아이디)를 입력 받는다

(1) 세부 기능 2

- 30개의 연애상황을 주고 그 상황에서 본인의 행동이 어떤 지에 대한 보기를 4개 주고 그에 따른 학생들의 답변을 입력 받는다.

2) 기능 2

- 학생들에게 입력 받은 데이터(답변)를 통해 5가지 항목에 대한 점수를 매긴다. (학생들의 답변마다 5가지 항목에 대한 점수 기여도가 다르다.)

3) 기능 3

- 5가지 항목을 통해 분류된 학생들 간의 연애관 및 성격의 유사성을 비교하여 퍼센트로 환산한다.

4) 기능 4

- 학생들 간의 연애관에 따른 상호보완성을 고려한 가산점을 부여한다.

5) 기능 5

- 환산한 퍼센트와 가산점을 합산하여 일정 수준 이상의 매칭 점수를 받은 학생들을 뽑아서 매칭을 시켜준다

3. 진척사항

1) 기능 구현

(1) 기능1(프로그램에 참여한 학생들의 정보를 입력받기)

- 입력 : 학생들의 이름, 인스타 아이디, 성별을 입력받음
- 설명 : 프로그램을 시작하기에 앞서 학생들의 개인정보를 입력받는 기능
- 적용된 배운 내용

1. Student라는 클래스를 형성하고 클래스로 student 객체를 만들.

2. 만든 student 객체들을 vector Profiles에 하나씩 저장하여 학생들의 개인정보들 및 프로그램에 필요한 데이터들을 객체로 묶어서 관리하고 참여하는 학생들의 수를 자유롭게 함.

3. 반복문 while문을 사용하여 학생들의 성별을 판단할 때 잘못된 정보를 입력했을 시 다시 입력하도록 하는 경고문을 작성함.

4. 조건문 if문을 사용하여 프로그램의 필요한 최소한의 학생 수가 모두 채워졌을 시 더 추가 할 것 인지를 물어봄.

5. push_back 함수를 사용하여 student 객체들을 Profiles 벡터에 넣는다.

-코드 스크린샷

The screenshot shows a C++ IDE with a file named `finalexam.cpp`. The code implements a while loop that repeatedly prompts the user for student information until the vector `Profiles` is full (10 students). The loop includes validation for gender (0 for male, 1 for female) and a warning message if an invalid character is entered. The console output shows the program running and accepting input.

```
19 vector<Student>Profiles;
20 int count = 0;
21 string strArray[6] = { "a1", "a2", "a3", "a4", "a5", "a6" };
22 while (true)
23 {
24     Student student;
25     cout << "참여자의 정보를 입력하시오 " << endl;
26     cout << "이름 : ";
27     cin >> student.name;
28     cout << "참여자의 인스타 아이디를 입력하시오 : ";
29     cin >> student.insta_ID;
30     while (true) {
31         cout << "참여자의 성별을 입력하시오 (남자일 경우 0, 여자일 경우 1) : ";
32         cin >> student.gender;
33         if (cin.fail()) {
34             cout << "잘못된 입력입니다. 숫자를 입력하세요." << endl;
35             cin.clear();
36             cin.ignore(INT_MAX, '\n');
37         }
38         else if (student.gender == 0 || student.gender == 1) {
39             break;
40         }
41         else {
42             cout << "잘못된 입력입니다. 0 또는 1을 입력하세요." << endl;
43         }
44     }
45     cout << endl;
46     count++;
47 }
48 if (count == 10) {
49     // ... (rest of the code)
50 }
```

The screenshot shows the same C++ IDE with the `finalexam.cpp` file. The code now includes a check at the beginning of the `main` function to see if the user wants to participate. If not, the program ends. If yes, it proceeds with the while loop for data entry. The console output shows the user entering 'Y' to participate.

```
1 #include<iostream>
2 #include<vector>
3 #include<string>
4 #include<iterator>
5 #include<program.h>
6 using namespace std;
7
8 int main() {
9     //프로그램 참여 신청하기
10    cout << "매장 프로그램에 참여하시겠습니까?(Y/N) : ";
11    char OK;
12    cin >> OK;
13    if (OK != 'y' && OK != 'Y') {
14        cout << "프로그램을 종료하겠습니다!!";
15        return 0;
16    }
17    else {
18        cout << "환영합니다!" << endl << endl;
19    }
20    vector<Student>Profiles;
21    int count = 0;
22    string strArray[6] = { "a1", "a2", "a3", "a4", "a5", "a6" };
23    while (true)
24    {
25        Student student;
26        cout << "참여자의 정보를 입력하시오 " << endl;
27        cout << "이름 : ";
28        cin >> student.name;
29        cout << "참여자의 인스타 아이디를 입력하시오 : ";
30        cin >> student.insta_ID;
31        while (true) {
32            // ... (rest of the code)
33        }
34        count++;
35    }
36    if (count == 10) {
37        // ... (rest of the code)
38    }
39 }
```

(2) 기능2(학생들에게 입력 받은 데이터를 통해 5가지 항목에 대한 점수를 매기고 계산한다.)

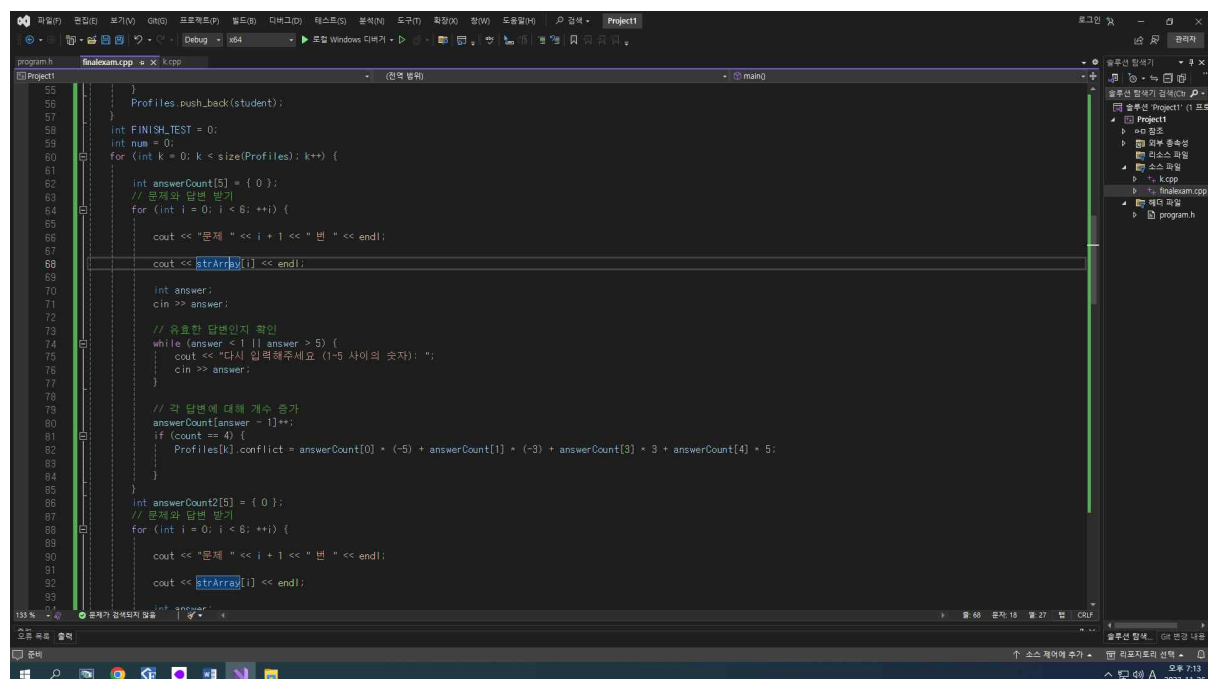
- 입력 : 학생들의 문제에 대한 답변을 입력받음
- 설명 : 각 문제에 대한 학생들의 답변을 입력받고 답변에 따른 점수를 매긴 후 학생들의 클래스에 저장하는 기능
- 적용된 배운 내용

1. 배열 answerCount를 선언 후 0으로 초기화 한 후 학생들의 답변을 answerCount라는 배열에 입력받아 사용함

2. 반복문 for문을 사용하여 배열 answerCount에 저장되어 있는 문제들을 하나씩 출력한다.

3. 반복문 while문을 사용하여 유효한 답변인지 확인 후 유효하지 않을 경우 다시 입력하도록 제안한다.

-코드 스크린 샷

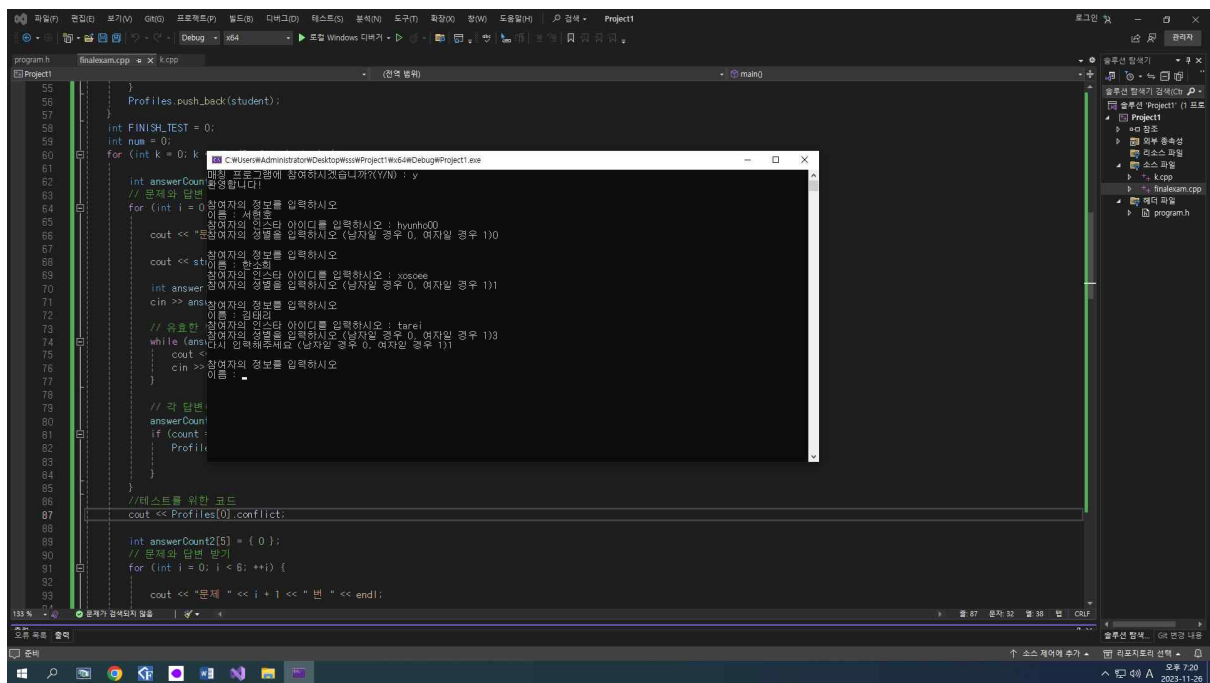
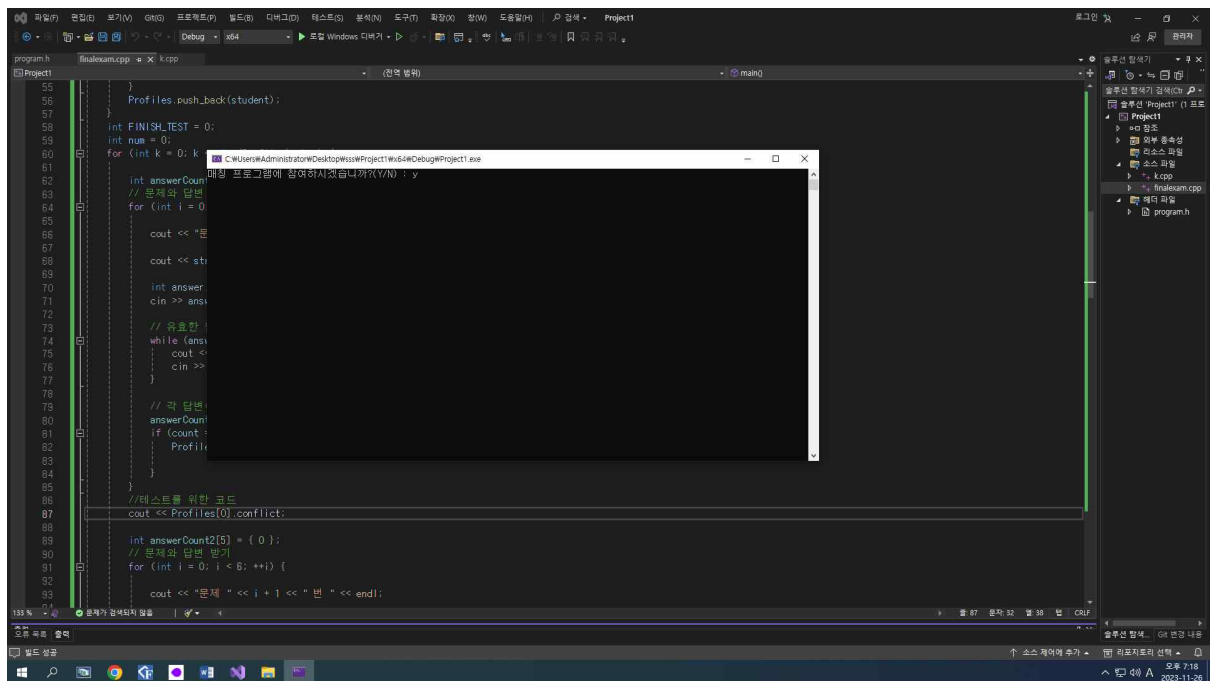


```
55 }
56 Profiles.push_back(student);
57 }
58 int FINISH_TEST = 0;
59 int num = 0;
60 for (int k = 0; k < size(Profiles); k++) {
61     int answerCount[5] = { 0 };
62     // 문제와 답변 받기
63     for (int i = 0; i < 6; ++i) {
64         cout << "문제 " << i + 1 << " 번 " << endl;
65         cout << strArray[i] << endl;
66         int answer;
67         cin >> answer;
68         // 유효한 답변인지 확인
69         while (answer < 1 || answer > 5) {
70             cout << "다시 입력해주세요 (1-5 사이의 숫자): ";
71             cin >> answer;
72         }
73         // 각 답변에 대해 개수 증가
74         answerCount[answer - 1]++;
75         if (count == 4) {
76             Profiles[k].conflict = answerCount[0] * (-5) + answerCount[1] * (-3) + answerCount[3] * 3 + answerCount[4] * 5;
77         }
78     }
79     int answerCount2[5] = { 0 };
80     // 문제와 답변 받기
81     for (int i = 0; i < 6; ++i) {
82         cout << "문제 " << i + 1 << " 번 " << endl;
83         cout << strArray[i] << endl;
```

2) 테스트 결과

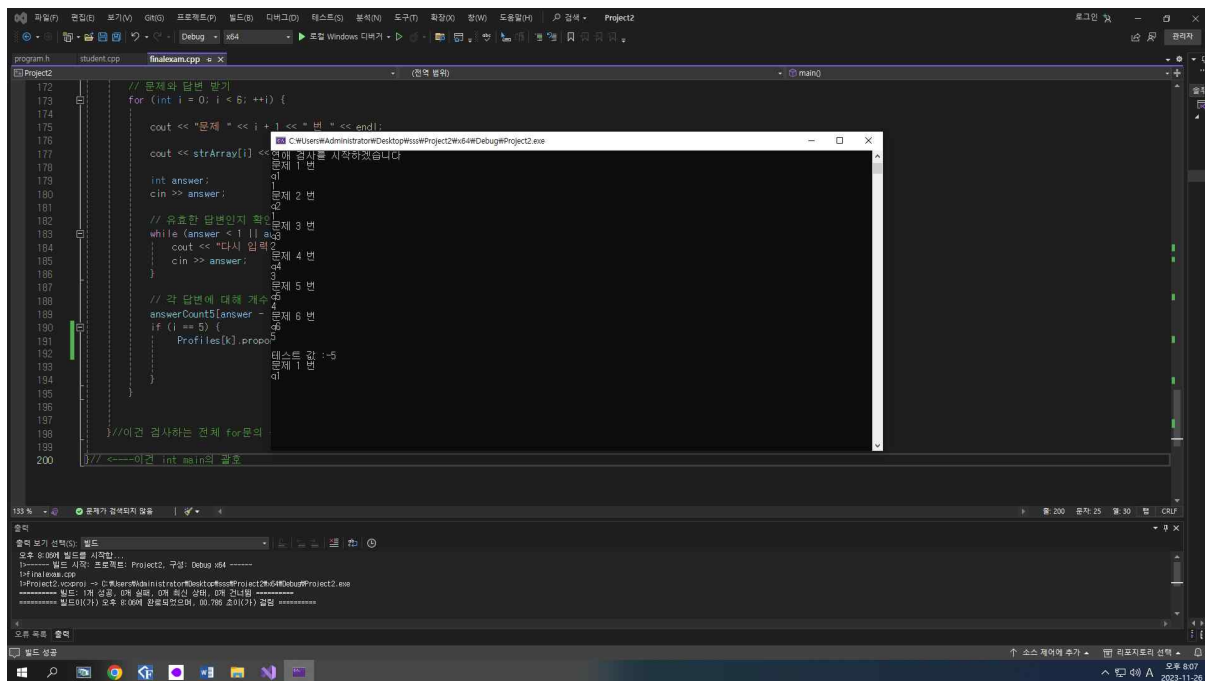
(1) 기능1(프로그램에 참여한 학생들의 정보를 입력받기)

- 설명 : 학생들의 개인정보를 입력 받고 잘못된 입력 값을 받은 경우 수정을 제안함
- 테스트 결과 스크린샷



(2) 기능2(학생들에게 입력 받은 데이터를 통해 5가지 항목에 대한 점수를 매기고 계산한다.)

- 설명 : 5가지 항목에 대한 점수를 매기고 여기서는 conflict 항목의 점수를 테스트 하는 코드를 추가하여 테스트 함 점수 (계산은 다음과 같음 : 1번 선택시 -5, 2번 선택시 -3, 3번 선택시 0, 4번 선택시 +3, 5번 선택시 +5)
- 테스트 결과 스크린샷



```
// 문제와 답변 받기
for (int i = 0; i < 5; ++i) {
    cout << "문제 " << i + 1 << " 번 " << endl;
    cout << strArray[i] << endl;
    int answer;
    cin >> answer;

    // 유효한 답변인지 확인
    while (answer < 1 || answer > 5) {
        cout << "다시 입력하세요" << endl;
        cin >> answer;
    }

    // 각 답변에 대한 점수
    answerCounts[answer - 1] += 1;
    if (i == 4) {
        Profiles[k].proposeScore = calculateScore(answerCounts);
    }
}

//이건 검사하는 전체 for문의 끝
//이건 int main의 끝
```

Console Output:

```
C:\Users\Administrator\Desktop\Project2\Debug\Project2.exe
문제 1 번
문제 2 번
문제 3 번
문제 4 번
문제 5 번
테스트 합 -5
문제 1 번
```

4. 계획 대비 변경 사항

1) 일정 순서 변경

- 이전

: 학생들에게 줄 문제를 선정하는 것을 11/24까지 완료하기로 함.

12/8 까지 점수 채점을 완료하기로 함

- 이후 : 점수 채점 코드를 먼저 짰 후 문제 선별을 12/8일 까지 하기로 함

- 사유 : 촉박한 기말시험 일정으로 시간이 더 많이 드는 것을 먼저 수행하기로 함.

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/10	11/17	11/24
제안서 작성		완료			
기능1	세부기능1		완료		
	세부기능2			완료	
기능2					완료
기능3					
기능4	세부기능1		진행 중		
기능5	세부기능2			----->	
날짜	12/1	12/8	12/15	12/22
기능1	세부기능1				
	세부기능2				
기능2		---->			
기능3		----->			
기능4		----->			
기능5		----->			